МИНОБРНАУКИ РОССИИ

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ

ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)

Кафедра САПР

ОТЧЕТ

по лабораторной работе №4

по дисциплине «Компьютерная графика»

Тема: Исследование алгоритмов отсечения отрезков и многоугольников окнами различного вида

Студенты гр. 1302 _____ Марзаева В.И.

_____ Новиков Г.В.

_____ Романова О.В.

Преподаватель _____ Колев Г.Ю.

Санкт-Петербург

2024

Цель работы

Обеспечить реализацию алгоритма отсечения массива произвольных отрезков заданным прямоугольным окном с использование алгоритма Коэна Сазерленда. Вначале следует вывести на экран сгенерированные отрезки полностью, а затем другим цветом или яркостью те, которые полностью или частично попадают в область окна.

Теоретическая часть программы

В лабораторной работе используется 2-х мерное отсечение (операции отсечения над отрезками). Отсечение необходимо, чтобы из обширной базы данных выделить отдельные элементы для вывода на дисплей.

Есть окно и множество отрезков, для которых нужно определить, какая часть отрезка попадает в окно:

1) виден полностью и его можно вывести в этом окне;
2) виден частично и тогда необходимо определить его видимую часть;
3) не виден полностью и его нужно исключить из дальнейшего рассмотрения.

Этапы алгоритма:

1 этап: выявить полностью видимые отрезки или тривиально невидимые в прямоугольном окне.

2 этап: у оставшихся (других) найти точки пересечения с границами окна, и определить видимую часть или убедиться, что он полностью невидим (нетривиально).

3 этап: вывести на экран.

Алгоритм Сазерленда-Коэна:

Известны размеры окна (параметры – значения координат окна): X левое, X правое, У верхнее, У нижнее и параметры концов отрезков.

Идея алгоритма сводится к переносу конечных точек, находящихся вне окна, на линии границ окна с отбрасыванием невидимых частей отрезка.

В начале используются 4-х битовые коды для концов каждого отрезка - – Л, П, Н, В. (левее, правее, ниже, выше окна):

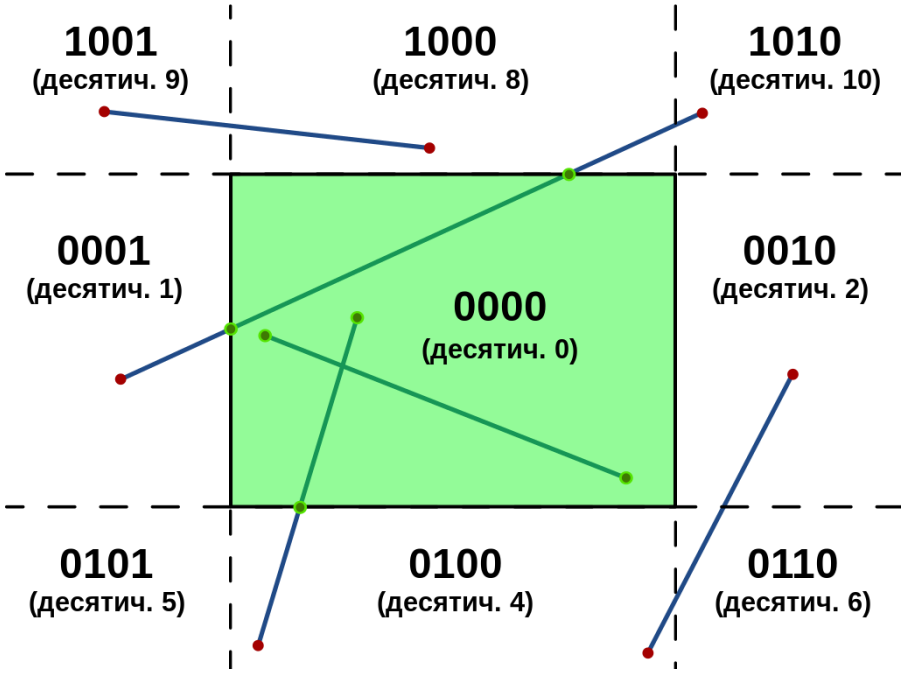

Рис. 1

Исходя из кодов:

Кн & Кк = 0 //полностью видимый отрезок

//т.е. код начала и конца тождественно = 0

Кн * Кк ≠ 0 // побитовое умножение ≠ 0, то отрезок тривиально не виден

Кн * Кк = 0 //побитовое умножение = 0, то отрезки могут быть не видны, либо частично видны

Алгоритм можно представить в следующем виде:

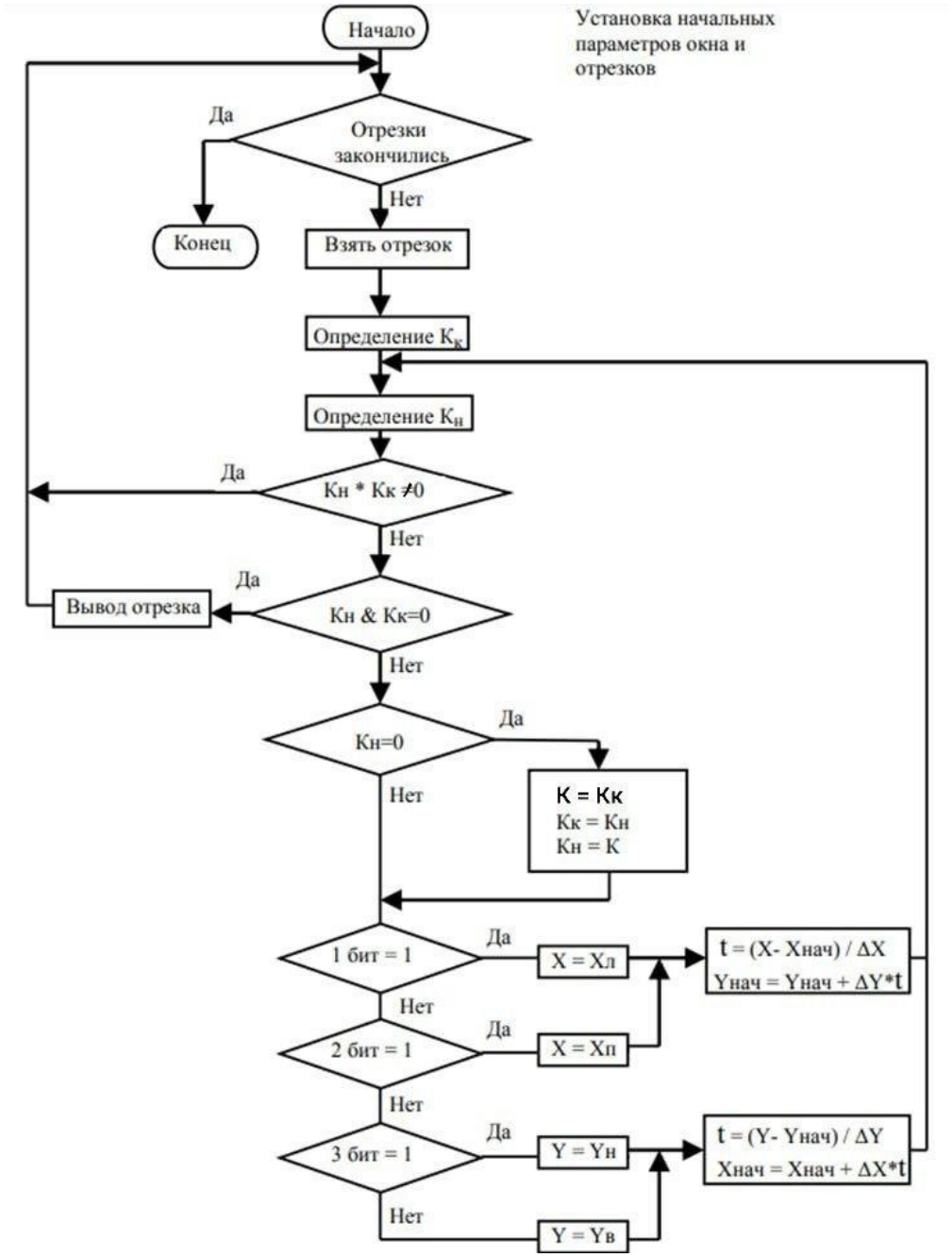

Рис. 2

Пример работы программы

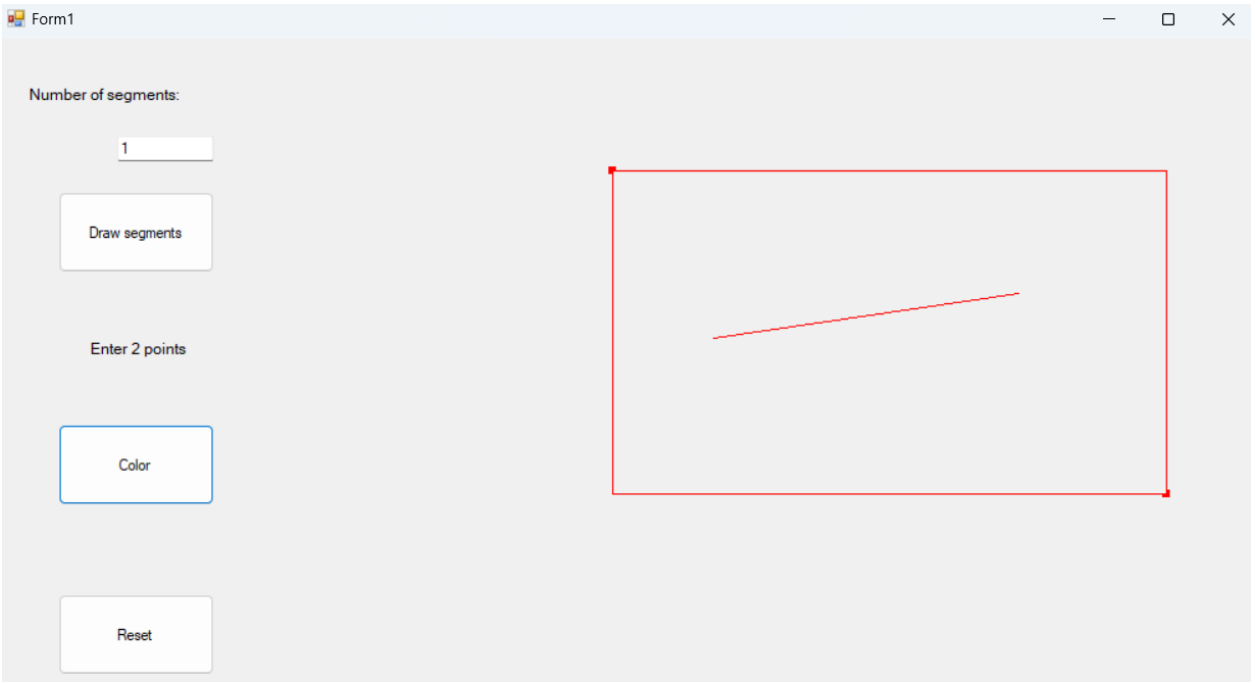

Рис. 3 – Пример работы программы с 1 отрезком

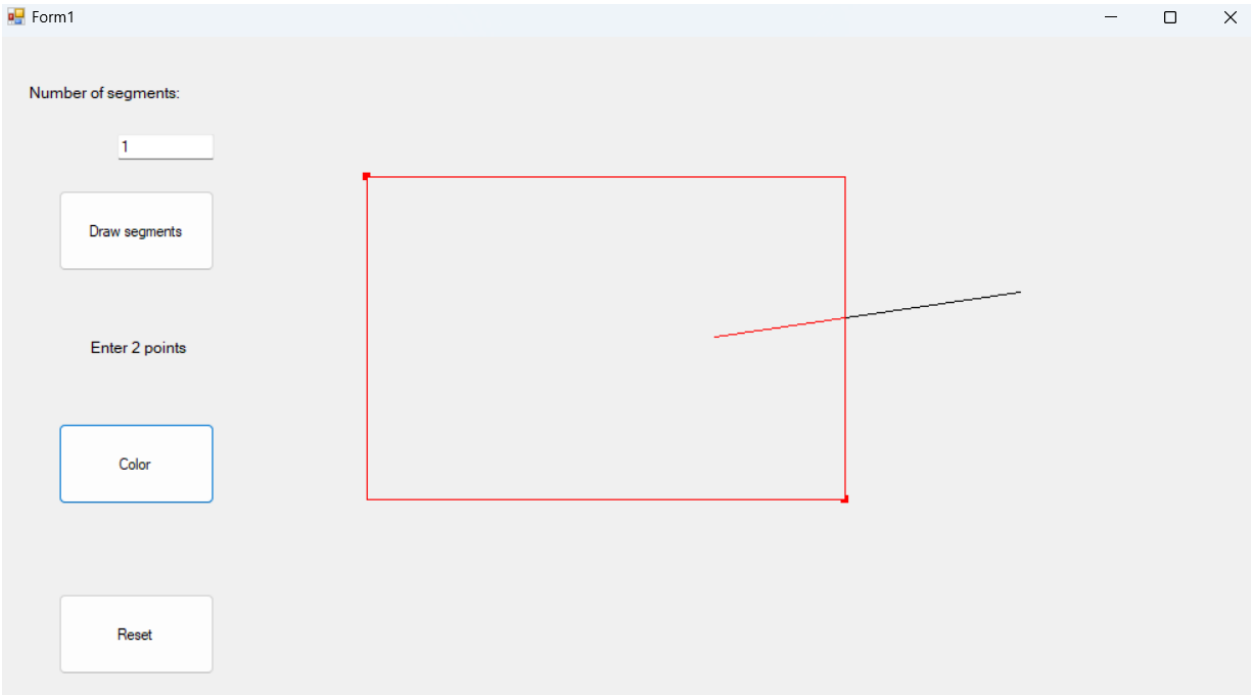

Рис. 4 – Пример работы программы с 1 отрезком

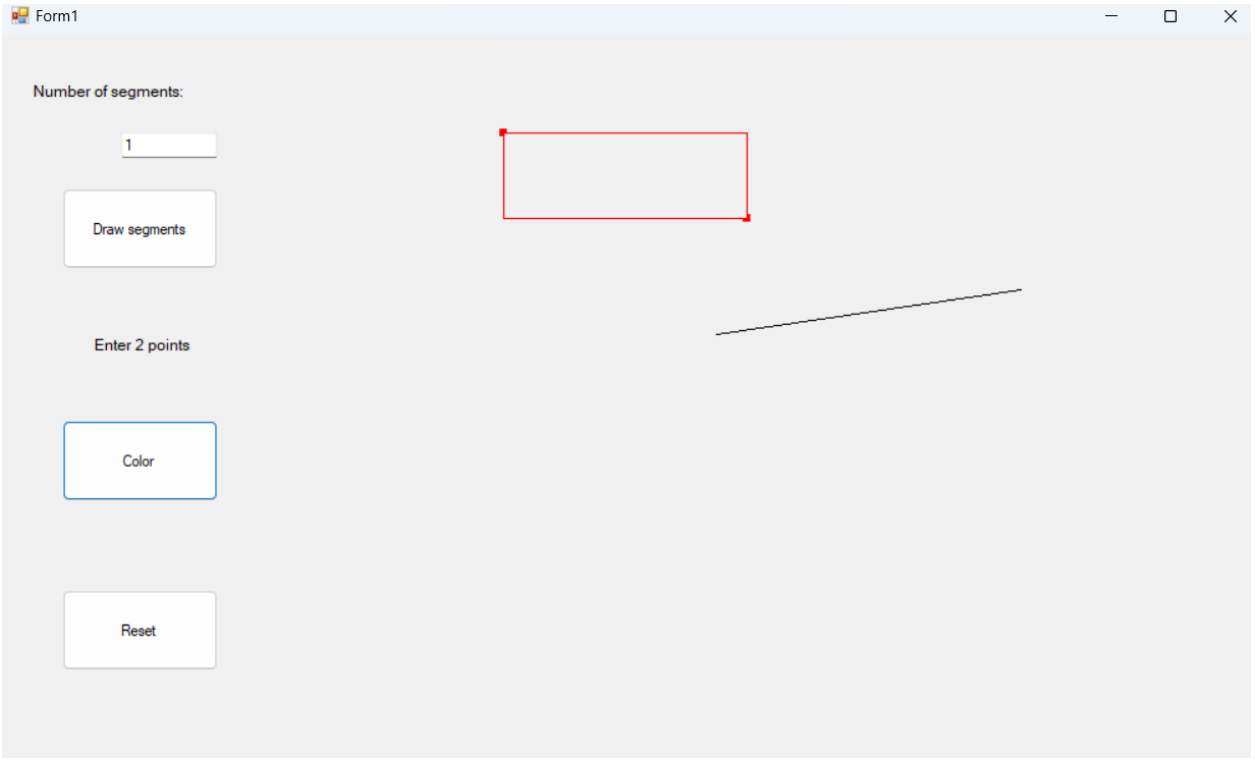

Рис. 5 – Пример работы программы с 1 отрезком

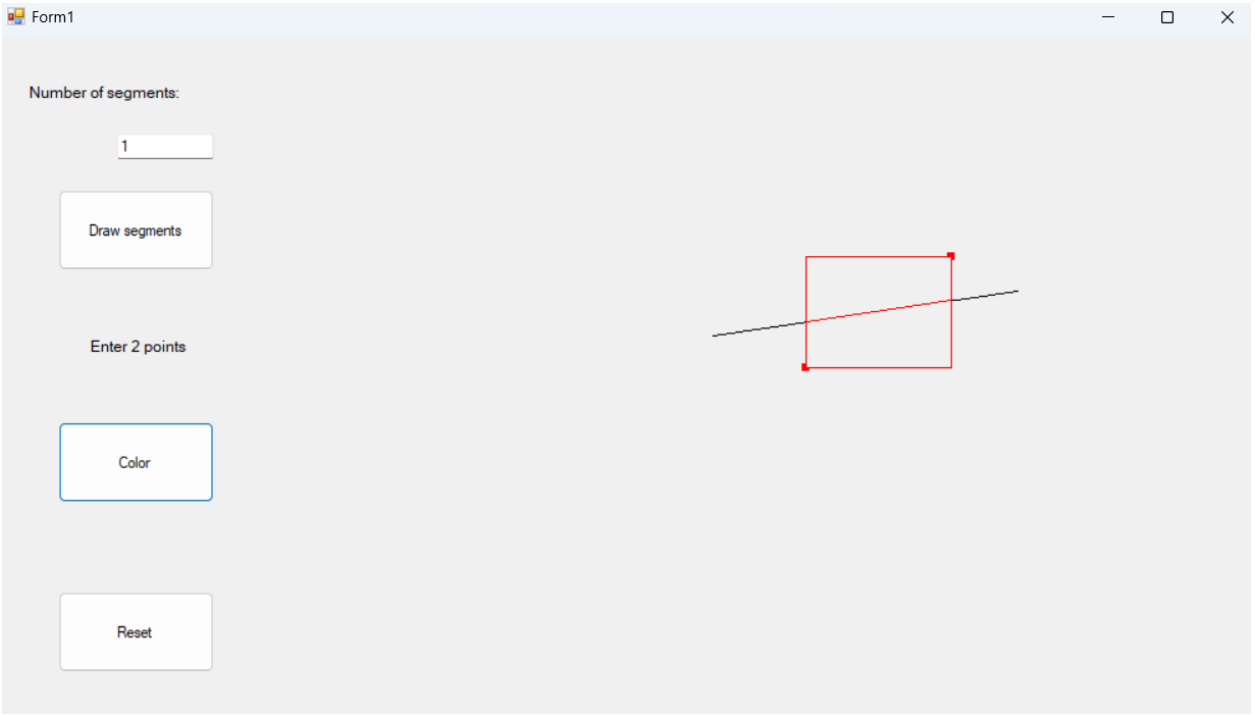

Рис. 6 – Пример работы программы с 1 отрезком

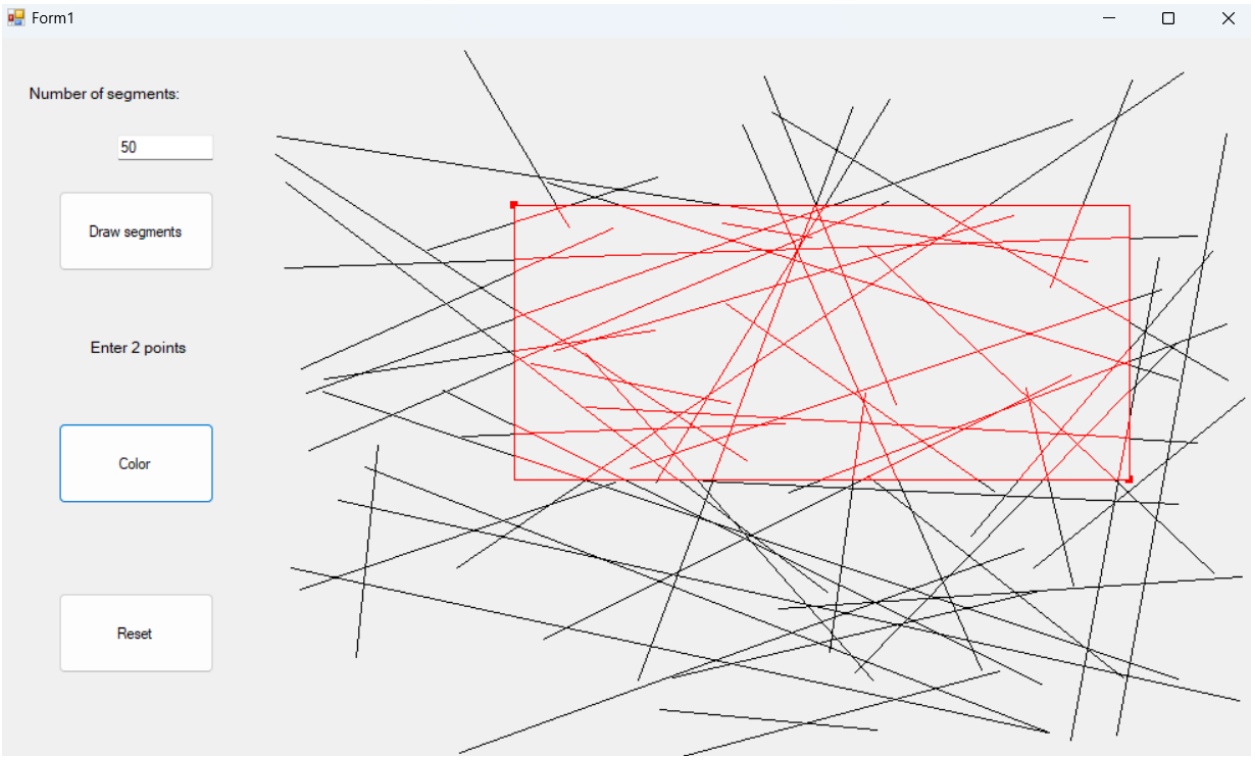

Рис. 7 – Пример работы программы с 50 отрезками

Код программы

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace lab4
{
   public partial class Form1 : Form
   {
      double x_point, y_point, x_point2, y_point2;
      double x_left, x_right, y_up, y_down;
      private Random random;
      private List<(int, int, int, int)> line = new List<(int, int, int, int)>();
      private List<(int, int, int, int)> redLine = new List<(int, int, int, int)>();
      bool isFirstClick = true;

      public Form1()
      {
         InitializeComponent();
         random = new Random();
      }

      private void Form1_Load(object sender, EventArgs e)
      {
         pictureBox1.Anchor = AnchorStyles.Top | AnchorStyles.Bottom | AnchorStyles.Left | AnchorStyles.Right;
```

```
    label1.Anchor = AnchorStyles.Top | AnchorStyles.Bottom | AnchorStyles.Left | AnchorStyles.Right;
    label2.Anchor = AnchorStyles.Top | AnchorStyles.Bottom | AnchorStyles.Left | AnchorStyles.Right;
    textBox1.Anchor = AnchorStyles.Top | AnchorStyles.Bottom | AnchorStyles.Left;

}

private void pictureBox1_Paint(object sender, PaintEventArgs e)
{
    Graphics g = e.Graphics;

}

private void button3_Click(object sender, EventArgs e)
{
    Pen pen = new Pen(Color.Red);
    SolidBrush brush = new SolidBrush(pictureBox1.BackColor);
    Graphics g = pictureBox1.CreateGraphics();

    redLine.Clear();

    for (int i = 0; i < line.Count; i++)
    {
        var l = line[i];
        string code_end = GetCode(l.Item3, l.Item4);
        string code_start = GetCode(l.Item1, l.Item2);
        string zero = "0000";
        bool res1 = false;
        bool res2 = false;
        char one = '1';

        if (code_start == zero && code_end == zero)
        {
            res1 = true;
        }

        int result = (Convert.ToInt32(code_start, 2)) & (Convert.ToInt32(code_end, 2));
        string res = Convert.ToString(result, 2).PadLeft(4, '0');

        if (res != zero)
        {
            res2 = true;
        }

        while (!res2 && !res1)
        {
            Algorithm(ref l, code_start, code_end, one, zero);

            code_start = GetCode(l.Item1, l.Item2);
            code_end = GetCode(l.Item3, l.Item4);

            if (code_start == zero && code_end == zero)
            {
                res1 = true;
            }

            result = (Convert.ToInt32(code_start, 2)) & (Convert.ToInt32(code_end, 2));
            res = Convert.ToString(result, 2).PadLeft(4, '0');

            if (res != zero)
            {
                res2 = true;
            }
        }
        if (res1)
        {
```

```
                redLine.Add((l.Item1, l.Item2, l.Item3, l.Item4));
            }
        }

        g.FillRectangle(brush, Convert.ToInt32(x_left) + 1, Convert.ToInt32(y_up) + 1, Convert.ToInt32(Math.Abs(x_right -
x_left)) - 1, Convert.ToInt32(Math.Abs(y_up - y_down)) - 1);
        for (int i = 0; i < redLine.Count; i++)
        {
            var rl = redLine[i];
            g.DrawLine(pen, rl.Item1, rl.Item2, rl.Item3, rl.Item4);
        }
    }

    private void pictureBox1_MouseClick(object sender, MouseEventArgs e)
    {
        Graphics g = pictureBox1.CreateGraphics();
        Pen pen = new Pen(Color.Red);

        if (isFirstClick)
        {
            g.Clear(BackColor);
            DrawLinesFromList(g);
            isFirstClick = false;

            x_point = e.X;
            y_point = e.Y;

            DrawFatPoint(g, (int)x_point, (int)y_point);

        }
        else
        {
            isFirstClick = true;

            x_point2 = e.X;
            y_point2 = e.Y;

            DrawFatPoint(g, (int)x_point2, (int)y_point2);

            g.DrawLine(pen, (int)x_point, (int)y_point, (int)x_point2, (int)y_point);
            g.DrawLine(pen, (int)x_point2, (int)y_point, (int)x_point2, (int)y_point2);
            g.DrawLine(pen, (int)x_point2, (int)y_point2, (int)x_point, (int)y_point2);
            g.DrawLine(pen, (int)x_point, (int)y_point2, (int)x_point, (int)y_point);

            if (x_point < x_point2)
            {
                x_left = x_point;
                x_right = x_point2;

                if (y_point < y_point2)
                {
                    y_up = y_point;
                    y_down = y_point2;
                }
                else
                {
                    y_up = y_point2;
                    y_down = y_point;
                }

            }
            else
            {
                x_left = x_point2;
                x_right = x_point;
```

```
            if (y_point > y_point2)
            {
                y_up = y_point2;
                y_down = y_point;
            }
            else
            {
                y_up = y_point;
                y_down = y_point2;
            }
        }
    }

}

private void button1_Click(object sender, EventArgs e)
{
    Graphics g = pictureBox1.CreateGraphics();
    Pen pen = new Pen(Color.Black);

    double num_segments = Convert.ToInt32(textBox1.Text);

    if (line.Count > 0)
    {
        line.Clear();
        g.Clear(BackColor);
    }

    for (int i = 0; i < num_segments; i++)
    {
        int x1 = random.Next(pictureBox1.Width);
        int y1 = random.Next(pictureBox1.Height);
        int x2 = random.Next(pictureBox1.Width);
        int y2 = random.Next(pictureBox1.Height);

        line.Add((x1, y1, x2, y2));
        DrawLinesFromList(g);
    }
}

private void button2_Click(object sender, EventArgs e)
{
    Graphics g = pictureBox1.CreateGraphics();
    g.Clear(BackColor);
    line.Clear();
}

private void pictureBox1_SizeChanged(object sender, EventArgs e)
{
    ((PictureBox)sender).Invalidate();
}

private void DrawFatPoint(Graphics g, int x, int y)
{
    SolidBrush redBrush = new SolidBrush(Color.Red);
    int fatness = 6;
    g.FillRectangle(redBrush, x - fatness / 2, y - fatness / 2, fatness, fatness);
}

private void DrawLinesFromList(Graphics g)
{
    Pen pen = new Pen(Color.Black);

    foreach (var l in line)
```

```
    {
        g.DrawLine(pen, l.Item1, l.Item2, l.Item3, l.Item4);
    }
}

private string GetCode(int x, int y)
{
    string code_str = "0000";
    StringBuilder sb = new StringBuilder(code_str);
    int indexToChange;
    char newChar = '1';

    if (x_left > x)
    {
        indexToChange = 0;
        sb[indexToChange] = newChar;
    }

    if (x_right < x)
    {
        indexToChange = 1;
        sb[indexToChange] = newChar;
    }

    if (y_down < y)
    {
        indexToChange = 2;
        sb[indexToChange] = newChar;
    }

    if (y_up > y)
    {
        indexToChange = 3;
        sb[indexToChange] = newChar;
    }

    string newCodeStr = sb.ToString();

    return newCodeStr;
}

private void Algorithm(ref (int, int, int, int) l, string code_start, string code_end, char one, string zero)
{
    string k = "";
    int it3 = l.Item3, it4 = l.Item4;
    double t;
    int newX, newY;

    if (code_start == zero)
    {
        k = code_start;
        code_start = code_end;
        code_end = k;
        l.Item3 = l.Item1;
        l.Item4 = l.Item2;
    }
    if (code_start[0] == one)
    {
        t = (x_left - l.Item1) / (it3 - l.Item1);

        newY = (int)(l.Item2 + (it4 - l.Item2) * t);

        l.Item2 = Math.Max(newY, 0);
        l.Item1 = (int)x_left;
    }
```

```
            else if (code_start[1] == one)
            {
                t = (x_right - l.Item1) / (it3 - l.Item1);

                newY = (int)(l.Item2 + (it4 - l.Item2) * t);

                l.Item2 = Math.Max(newY, 0);
                l.Item1 = (int)x_right;
            }
            else if (code_start[2] == one)
            {
                t = (y_down - l.Item2) / (it4 - l.Item2);

                newX = (int)(l.Item1 + (it3 - l.Item1) * t);

                l.Item1 = Math.Max(newX, 0);
                l.Item2 = (int)y_down;
            }
            else
            {
                t = (y_up - l.Item2) / (it4 - l.Item2);

                newX = (int)(l.Item1 + (it3 - l.Item1) * t);

                l.Item1 = Math.Max(newX, 0);
                l.Item2 = (int)y_up;
            }

        }
    }
}
```

Выводы

В данной работе с помощью Windows Forms на C# была реализована программа, которая выводит отрезки на экран, потом пользователь вводит координаты прямоугольного окна, внутри которого отрезки выделяются другим цветом. Для отсечения был использован алгоритм Коэна-Сазерленда.